41

Attorney's Docket No.: 042390.P6447

AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application for:

Michael Chow, et al.

Application No.: 09/505,949

Filed: February 15, 2000

For: **METHOD AND APPARATUS FOR ACHIEVING ARCHITECTURAL CORRECTNESS IN A MULTI-MODE PROCESSOR PROVIDING FLOATING-POINT SUPPORT**

Examiner: Li, Aimee J.

Art Group: 2183

SUPPLEMENTAL APPEAL BRIEF

Mail Stop Appeal Brief - Patent
Commissioner for Patents
P. O. 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicants request reinstatement of Appeal pursuant to 37 C.F.R. §1.193(b)(1)(ii). Applicants the following Supplemental Appeal Brief, in triplicate, pursuant to 37 C.F.R. §1.192(c) for consideration by the Board of Patent Appeals and Interferences. Should any charges be required, please charge any additional amount due or credit any overpayment to deposit Account No. 02-2666.

TABLE OF CONTENTS

		Page
I.	REAL PARTY IN INTEREST	2
II.	RELATED APPEALS AND INTERFERENCES	2
III.	STATUS OF CLAIMS	2
IV.	STATUS OF AMENDMENTS	2
V.	SUMMARY OF THE INVENTION	2
VI.	ISSUES	3
VII.	GROUPING OF CLAIMS	3
VIII.	ARGUMENT	4
A.	Overview of the Invention and Cited References	4
1.	Overview of Invention	4
2.	Overview of Mohammed Reference	5
3.	Overview of Blomgren Reference	6
4.	Overview of Loper Reference	7
B.	Group I: Rejection of Claim 5 Under 35 U.S.C. §112, Second Paragraph	8
C.	Group II: Rejection of Claims 1, 2, 4, 5, 7, 8 and 18 as Obvious Over Mohammed in View of Blomgren	9
1.	Errors of Law and Fact in the Rejection	9
2.	Specific Limitations Not Described in the Prior Art	10
3.	Explanation Why Such Limitations Render the Claims Non-obvious over the Prior Art	10
D.	Group III: Rejection of Claim 3 As Obvious Over Mohammed in View of Blomgren and Further in View of Loper	12
1.	Errors of Law and Fact in the Rejection	12
2.	Specific Limitations Not Described in the Prior Art	13
3.	Explanation Why Such Limitations Render the Claims Non-obvious over the Prior Art	13
E.	Group IV: Rejection of Claim 6 As Obvious Over Mohammed in View of Blomgren and Further in View of Loper	14
1.	Errors of Law and Fact in the Rejection	14
2.	Specific Limitations Not Described in the Prior Art	14
3.	Explanation Why Such Limitations Render the Claims Non-obvious over the Prior Art	14
F.	Group V: Rejection of Claim 9 as Obvious over Mohammed in View of Blomgren	15
1.	Errors of Law and Fact in the Rejection	15
2.	Specific Limitations Not Described in the Prior Art	15
3.	Explanation Why Such Limitations Render the Claims Non-obvious over the Prior Art	15

Page

G. Group VI: Rejection of Claims 12, 13, 15, 16 and 19 As Obvious Over Mohammed in View of Blomgren 16
1. Errors of Law and Fact in the Rejection 16
2. Specific Limitations Not Described in the Prior Art 17
3. Explanation Why Such Limitations Render the Claims Non-obvious over the Prior Art 17

H. Group VII: Rejections of Claim 10 18

1. Rejection of Claim 10 As Obvious Over Mohammed in View of Blomgren 18
a. Errors of Law and Fact in the Rejection 18
b. Specific Limitations Not Described in the Prior Art 19
c. Explanation Why Such Limitations Render the Claims Non-obvious over the Prior Art 19

2. Rejection of Claim 10 As Anticipated by Loper 19
a. Errors of Law and Fact in the Rejection 20
b. Specific Limitations Not Described in the Prior Art 20
c. Explanation Why Such Limitations Render the Claims Unanticipated by the Prior Art 21

I. Group VIII: Rejection of Claim 14 as Anticipated by Loper 21
1. Errors of Law and Fact in the Rejection 21
2. Specific Limitations Not Described in the Prior Art 22
3. Explanation Why Such Limitations Render the Claims Unanticipated by the Prior Art 22

J. Group IX: Rejection of Claim 17 as Obvious over Mohammed in View of Blomgren 23
1. Errors of Law and Fact in the Rejection 23
2. Specific Limitations Not Described in the Prior Art 23
3. Explanation Why Such Limitations Render the Claims Non-Obvious over the Prior Art 23

IX. CONCLUSION AND RELIEF 24

X. APPENDIX 25

I. REAL PARTY IN INTEREST

Michael Chow, Elango Ganesan, John William Phillips and Nazar Abbas Zaidi, the parties named in the caption, transferred their rights to that which is disclosed in the subject application through an assignment recorded on March 20, 2000 (010760/0706) in the patent application to Intel Corporation, of Santa Clara, California. Thus, as the owner at the time the brief is being filed, Intel Corporation, of Santa Clara, California is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences which will affect or be affected by the outcome of this appeal.

III. STATUS OF CLAIMS

Claims 1-19 are pending and rejected in this application. Applicants hereby appeal the rejection of all pending claims.

IV. STATUS OF AMENDMENTS

The claims are amended in accordance with the Response Amendment filed on August 8, 2003, wherein Claim 18 was amended. The claim amendments requested in the Response Amendment filed on January 15, 2004 regarding Claims 10 and 19 were not entered.

V. SUMMARY OF THE INVENTION

The present invention describes a multi-mode processor to process instructions from a first instruction set architecture (ISA) having a first word size and to process instructions from a second ISA having a second word size where the second word size of the second ISA is different than the first word size of the first ISA. As described at page 7 of Applicants' specification,

> In one embodiment, a processor is capable of operating in two modes. The first and second modes are a 32 bit word ISA and a 64 bit word ISA, respectively. More specifically, the first mode is IA-32 mode, in which the processor emulates a 32 bit word Intel architecture (IA). . . . [T]he second mode is IA-64, which implements what is known as the IA-64 ISA. (pg. 7, lines 1-16.)

Accordingly, in one embodiment, the multi-mode processor provides backward compatibility or legacy support for a 32 bit word (legacy) ISA, as well as support for a 64–bit word (current) ISA. As described, the term "word size", as known to those skilled in the art, refers to the number of bits that a CPU can process at one time. In one embodiment, the processor includes floating point registers and floating point units, which are shared between a legacy ISA engine and a current ISA engine, as illustrated with reference to FIG. 1. In a computer's arithmetic logic unit (ALU), there sometimes exists input encodings, which the machine must interpret as tokens, which

require special handling. To provide support for such encodings, the current ISA includes a special FP encoding ("NaTVal"). When such a value is detected, the processor known value causes a floating point unit to ignore the requested operation and propagate the NaTVal as output, typically causing the processor to later request the data and/or operation non-speculatively when it is needed. (*See* pg. 8, lines 2-19.)

Unfortunately, the legacy ISA does not support NaTVal tokens. However, because the floating point registers and floating point units are shared between the legacy ISA engine and current ISA engine of the multi-mode processor, in one embodiment, as illustrated with reference to FIG. 2, preprocessing hardware 162 detects whether a NaTVal token (or other token and special values) is present in input operands. When a NaTVal token or other special token is detected by preprocessing hardware 166, depending on what mode the processor is in, values other than a true arithmetic result are prepared by post-processing hardware 166. (*See* pg. 11, lines 4-17.)

Operation of the multi-mode processor is illustrated with reference to FIG. 3. Representatively, preprocessing hardware analyzes input operands to detect which input operands must be interpreted as tokens instead of being fed to the arithmetic unit. Also, where a result is produced, post-processing hardware replaces the arithmetic result with the expected result given the special input tokens. However, the capability to inject the special result is turned on or off depending on whether the multi-mode processor is processing instructions from the legacy ISA or the current ISA which supports NaTVal tokens.

VI. ISSUES

Is Claim 5 insufficiently definite to satisfy the requirements of 35 U.S.C. §112, second paragraph, for lacking antecedent basis?

Are Claims 1, 2, 4, 5, 7-13 and 15-19 unpatentable under 35 U.S.C. §103(a) as being obvious over Mohammed, U.S. Patent No. 6,366,998 ("Mohammed") in view of Blomgren, U.S. Patent No. 5,884,057 ("Blomgren")?

Are Claims 10 and 14 unpatentable under 35 U.S.C. §102(b) as being anticipated by Loper, U.S. Patent No. 5,611,063 ("Loper")?

Are Claims 3 and 6 unpatentable under 35 U.S.C. §103(a) as being obvious over Mohammed in view of Blomgren, as applied to Claim 1, and further in view of Loper?

VII. GROUPING OF CLAIMS

Applicants submit that claims do not stand or fall together. Accordingly, Applicants group the claims as follows:

Group I	Claim 5 for purposes of the rejection under §112
Group II	Claims 1, 2, 4, 5, 7, 8 and 18 (Claim 5 is included in Group II for all other rejections)

Group III	Claim 3
Group IV	Claim 6
Group V	Claim 9
Group VI	Claims 12, 13, 15, 16 and 19
Group VII	Claim 10
Group VIII	Claim 14
Group IX	Claim 17.

Applicants contend that all of the pending claims do not stand or fall together for the following reasons:

Claim 5 of Group I is separately grouped from the claims of Group II for purposes of the rejection of Claim under 35 U.S.C. §112, second paragraph. Claims 5 is included in Group II for all other rejections.

The claims in Group II specifically recite

> a first instruction set engine to process instructions from a first instruction set architecture (ISA) having a first word size and a second instruction set engine to process instructions from a second ISA having a second word size, the second word size being different than the first word size. (Emphasis added.)

The claims in Group III require preprocessing hardware and post-processing hardware to produce an output depending on a processor mode and whether an input includes a token.

The claims in Group IV require that a token representing a "not a thing value" (NaTVal) defines an unsuccessful speculative load request.

The claims of Group V require a 32-bit word ISA mode and a 64-bit word ISA mode.

The claims of Group VI require pre-processing and post-processing to produce an output depending on one of a plurality of processor modes and whether an input includes a token.

The claims of Group VII require pre-processing and post-processing to produce an output depending on a processor mode and whether an input includes a token.

The claims of Group VIII require that a token representing a "not a thing value" (NaTVal) defines an unsuccessful speculative load request.

The claims of Group IX require a 32-bit word ISA mode and a 64-bit word ISA mode.

In addition, other limitations require the claims to be grouped as indicated. Applicants will argue why each of these groups of claims should be allowed.

VIII. ARGUMENT

A. Overview of the Invention and Cited References

1. Overview of Invention

The present invention is directed to a multi-mode processor. Distinctive features of the multi-mode processor include:

(i) a first instruction set engine to process instructions from a first instruction set architecture (ISA) having a first word size;

(ii) a second instruction set engine to process instructions from a second ISA having a second word size, the second word size being different than the first word size; and

(iii) preprocessing and post-processing to produce an output depending on a mode of the multi-mode processor and whether an input operand includes a token.

The multi-mode processor supports a legacy ISA having, for example, a 32 bit word size and a current ISA, having a 64 bit word size. An arithmetic logic unit (ALU) of the multi-mode processor interprets certain input encodings as tokens that require special handling by the ALU. To provide support for such encodings, the current ISA includes a special floating point (FP) encoding ("NaTVal"). Unfortunately, the legacy ISA does not support NaTVal tokens. However, because the FP registers and FP units are shared between a legacy ISA engine and a current ISA engine of the multi-mode processor, multi-mode processor includes preprocessing and post-processing hardware.

As illustrated with reference to FIG. 2 of Applicant's specification, preprocessing hardware 162 detects whether a NaTVal token (or other special token and special values) are present in input operands. When a NaTVal token or other special token is detected by the preprocessing hardware 166, depending on what mode the processor is in, values other than the true arithmetic result are prepared by post-processing hardware 166. (*See* pg. 11, lines 4-17.) Accordingly, the capability to inject the special result is turned on or off, depending on whether the multi-mode processor is processing instructions from the legacy ISA or the current ISA, which supports NaTVal tokens.

2. Overview of Mohammed Reference

Mohammed describes:

A hybrid VLIW-SIMD programming model for use with a digital signal processor (DSP). (*See*, col. 3, lines 51-53.)

As further described by Mohammed:

[T]he ISA of the hybrid VLIW-SIMD programming model provides for instruction packets having different bit lengths . . . Preferably, instruction packets are 256 bits in length, as shown in FIG. 1, but this length is exemplary and not so limited. The instructions within the instruction packets may be of various lengths, preferably 32-bit instructions (i.e., long format) and 16-bit instructions (i.e., short format). (*See*, col. 5, lines 52-61.)

As also further described with Mohammed:

> Preferably, each instruction packet contains a number of mode bits to identify whether each instruction contained in the packet is in the long format or short format. A mode field may contain a number of mode bits. For example, a mode field of a 256-bit instruction packet is preferably a 14-bit field containing seven sub-fields, each two bits wide. A first bit value can be set to identify a long format instruction, and a second bit value can be set to identify a short form instruction. (*See*, col. 6, lines 21-29.)

The mode field of the 256-bit instruction packets, which may contain a 14-bit field containing seven sub-fields, each two bits wide, refers to whether the various instructions within a packet are either a long format or a short format. In other words, a field within an instruction packet to indicate whether instructions within the packet are in a long format or short format.

Although Mohammed teaches processing of short format and long format instructions, Mohammed provides no teachings or suggestions with regards to first and second ISAs to share the floating point registers. Specifically, as indicated above, the scheduler or scoreboard unit determines which functional units are available for executing the instructions. As described within Mohammed:

> Instructions 104-116 of the hybrid VLIW-SIMD DSP 100 are preferably received by a scheduler or scoreboard unit 120 which then determines which functional units are available for executing the instructions. Instructions 104-116 are then broadcast [to] datapath units ("DPUs") 122, each of which typically includes a plurality of functional units or processing elements . . . The functional units within DPUs 122 may be dynamically reconfigured to execute the variety of operations that may be required by the instructions 104-116. That is, in one cycle, one add unit may be used in connection with another add unit to execute a 64-bit vector add function. However, in another cycle, the same add unit may be used to execute a 16-bit scalar operation. (*See*, col. 4, lines 48-65.) (Emphasis added.)

The dynamic reconfiguration capability of the functional units enables Mohammed to provide a single ISA for a hybrid VLIW-SIMD programming model for instruction packets having different bit lengths. Hence, the teachings or Mohammed are specifically limited to a single ISA, which uses a mode field within received instruction packets to determine whether instructions within received packets are a long or short format. Based on the instruction format, the functional units are dynamically reconfigured to execute such instructions.

3. Overview of Blomgren Reference

Blomgren teaches a mechanism for rapid reconfiguration of pipeline alignment between a pipeline optimized for RISC instructions and one optimized for CISC instructions with the use of muxes and mode registers. (Abstract, and also as depicted with reference to Blomgren's Fig. 2.) As indicated in Blomgren:

The RISC and CISC instruction sets have independent encoding of instructions to opcodes. While both sets have ADD operations, the opcode number which encodes the ADD operation is different for the two instruction sets. In fact, the size and location of the opcode field in the instruction word is also different for the two instruction sets. Thus two instruction decoders are used for the two instruction sets – a RISC decoder 36 and a CISC decoder 32. (Col. 6, lines 37-44.) [Emphasis added.]

To this end, Blomgren describes various mechanisms for aligning the CISC and RISC pipelines, as depicted with reference to Figs. 2-3 and 5-7.

As illustrated with reference to FIG. 4:

the CISC instruction decoder 32 detects these emulated instructions and signals from unknown opcode over line 40 to mode control logic 30. In response, the mode control logic 30 sets RISC bit-60 in register 38 and loads the instruction pointer with the address of the emulation routine in memory. Once the emulation routine is complete, an RISC instruction causes the mode register 38 to be reset to CISC mode and the instruction pointer updated to point to the following CISC instruction. The CISC program continues with the following instruction unaware that the instruction was emulated with RISC instructions. (*See* cols. 6-7 lines 61–10.) (Emphasis added.)

Accordingly, Blomgren teaches the emulation of complex CISC instructions using a plurality of RISC instructions, which are referred to as "emulated instructions".

The instruction decoder 32 is responsible for detecting such instructions, which results in the setting of an RISC mode and loading of the emulation routine in memory, such that following execution, the mode register 38 is reset to CISC mode and the instruction pointer is updated to point to the following CISC instruction. (Col. 7, lines 1-10.) (Emphasis added.)

In other words, Blomgren teaches the detection of an emulated CISC instruction and switches to a RISC mode when such an instruction is detected. The switch to the RISC mode results in the loading of various RISC instructions to perform the CISC instruction. Once completed, the mode is switched to the CISC mode to achieve rapid reconfiguration of pipeline alignment between a pipeline optimized for RISC instructions and a pipeline optimized for CISC instructions.

4. Overview of Loper Reference

Loper describes a method for executing speculative load instructions in high performance processors; specifically:

[W]hen a speculative load instruction for which the data is not stored in a data cache is encountered, a bit within an enable speculative load table which is associated with that particular speculative load instruction is read in order to determine a state of the bit. If the associated bit is in a first state, data for the speculative load instruction is requested from a system bus and further execution of the speculative load instruction is then suspended to wait for control signals from a branch processing unit. If the associated bit is in the second state, the execution of

the speculative load instruction is immediately suspended to wait for control signals from the branch processing unit. . . . In this manner, the displacement of useful data in the data cache due to wrongful execution of the speculative load instruction is avoided. (*See*, col. 2, lines 2-23.) (Emphasis added.)

In other words, the teachings of Loper are strictly directed to prohibiting the displacement of:

> essential data stored in a data cache by some irrelevant data obtained from the system bus because of a wrongful execution of a speculative load instruction caused by misprediction. (*See*, col. 1, lines 44-47.)

FIG. 2A of Loper illustrates the enable speculative load table, which contains a bit associated with various speculative load instructions. Specifically:

> With reference now to FIG. 2A, there depicted a block diagram of an Enable Speculative Load (ESL) table for facilitating the selective execution of speculative load instructions . . . ESL table 60 is a small memory location, preferably within processor 10, indexed by a lower portion of a load instruction address. (*See*, col. 4, lines 34-40.)

The teachings of Loper are specifically limited to either requesting data for a speculative load instruction or immediately suspending the speculative load instruction according to the state of a bit in the ESA table as shown in FIGS. 2A and 2B associated with the speculative load instruction.

B. Group I: Rejection of Claim 5 Under 35 U.S.C. §112, Second Paragraph

In the Office Action mailed on June 30, 2004, the Examiner rejected Claim 5 under 35 U.S.C. §112, second paragraph, for lacking antecedent basis for the recitation of "the token" in line 2 of Claim 5. Applicants respectfully submit that Claim 5 provides sufficient antecedent basis for the limitation "the token" in line 2 of Claim 5. Specifically, Claim 5 recites "wherein the input may contain a token" and subsequently refers to "a token" as "the token being an 82-bit processor known value."

Accordingly, Applicants respectfully submit that the initial reference to "a token" in lines 1 and 2 of Claim 5 provides antecedent basis for the reference to "the token" in line 2 of Claim 5 alluded to by the Examiner. Accordingly, Applicants respectfully submit the 35 U.S.C. §112, second paragraph, rejection of Claim 5 is in error since Claim 5 particularly points out and distinctly claims subject matter, which Applicants regard as the invention. Applicants request the Board overturn this rejection.

C. Group II: Rejection of Claims 1, 2, 4, 5, 7, 8 and 18 as Obvious Over Mohammed in View of Blomgren

The Examiner rejected all pending claims, including Claims 1, 2, 4, 5, 7, 8, 17 and 18 of Group I under 35 U.S.C. §103(a) as being obvious over Mohammed in view of Blomgren.

1. Errors of Law and Fact in the Rejection

The Federal Circuit Court of Appeals in In re Rijckaert, 9 F.3d 1531, 28 U.S.P.Q. 2d 1955 (Fed. Cir. 1993) held that:

> In rejecting claims under 35 U.S.C. § 103, the examiner bears the initial burden of presenting a *prima facie* case of obviousness. . . . "A *prima facie* case of obviousness is established when the teaching from the prior art itself would appear to have suggested the claimed subject matter to a person of ordinary skill in the art.". . . If the examiner fails to establish a *prima facie* case, the rejection is improper and will be overturned. (Emphasis added.) 9 F.3d at 1532, 28 U.S.P.Q. 2d at 1956.

Although the Examiner has rejected Claims 1, 2, 4, 5, 7, 8 and 18 of Group II as obvious over Mohammed in view of Blomgren, the combination fails to teach the claimed subject matter. Furthermore, Applicants respectfully submit that the features of the claims of Group II could only be arrived at through inappropriate hindsight.

The dynamic reconfiguration capability of the functional units taught by Mohammed enables a single ISA for a hybrid VLIW-SIMD programming model for instruction packets having different bit lengths. Hence, Applicants respectfully submit that the teachings or Mohammed are specifically limited to a single ISA, which uses a mode field within received instruction packets to determine whether instructions within received packets are a long or short format. Based on the instruction format, the functional units are dynamically reconfigured to execute such instructions.

Therefore, since Mohammed is specifically limited to a single ISA, Mohammed cannot be modified according to Blomgren to share floating point registers between first and second ISAs, as recited by the claims of Group II. Furthermore, Applicants respectfully submit that the Examiner fails to illustrate a teaching or suggestion for combining the teachings of Mohammed in view of Blomgren.

It is well established that obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention absent the teaching or suggestion supporting such combination. ACS Hospital Sys., Inc. v. Montefiore Hospital, 732 F.2d. 1572, 1577, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984). Also, one cannot find obviousness through hindsight to construct a claimed invention from elements of the prior art. In re Warner, 379 F.2d 1011, 1016, 154 U.S.P.Q. 173, 177 (C.C.P.A. 1967).

Hence, a *prima facie* case of obviousness of the claims of Group II has not been established and the rejection of Claims 1, 2, 4, 5, 7, 8 and 18 is erroneous and should be overturned.

2. Specific Limitations Not Described in the Prior Art

Each of the independent claims of Group II require:

(1) a mode identifier;
(2) a plurality of floating-point registers shared by the first instruction set engine and the second instruction set engine; and
(3) a floating-point unit coupled to the floating-point registers, the floating-point unit processing an input responsive to the mode identifier to produce an output. (Emphasis added.)

3. Explanation Why Such Limitations Render the Claims Non-obvious over the Prior Art

The Applicants claim a multi-mode processor that includes a first instruction set engine to process instructions from a first instruction set architecture ISA having a first word size. The multi-mode processor also includes a second instruction set engine to process instructions from a second ISA having a second word size, the second word size being different than the first word size. As described at pg. 7 of Applicants' specification:

> In one embodiment, a processor is capable of operating in two modes. The first and second modes are a 32 bit word ISA and a 64 bit word ISA, respectively. More specifically, the first mode is IA-32 mode, in which a processor emulates a 32 bit word Intel Architecture (IA) known as IA-32 ISA . . . [T]he second mode is a IA-64, which implements what is known as the IA-64 ISA. (pg. 7, lines 1-16.) (Emphasis added.)

Accordingly, in one embodiment, the multi-mode processor provides backward compatibility or legacy support for a 32 bit word (legacy ISA), as well as support for a 64 bit word (current) ISA.

According to the Examiner, Mohammed teaches a mode identifier at col. 4, lines 28–47; col. 5, line 39 to col. 6, line 34; col. 8, lines 13-63; col. 9, lines 37-44; and FIGS. 1-8. (*See*, page 4 of Office Action mailed June 30, 2004.) Applicants respectfully disagree with the Examiner's contention. Applicants respectfully submit that the claims of Group II recite a mode identifier of a processor. The mode identifier is analogous to a global mode of the processor and is therefore distinct from a state of the various instructions executed by the processor.

Applicants respectfully submit that the mode field of the 256-bit instruction packets, which may contain a 14-bit field containing seven sub-fields, each two bits wide, refers to whether the various instructions within a packet are either a long format or a short format. In other words, a field within an instruction packet to indicate whether instructions within the packet are in a long

format or short format does not teach or suggest a mode identifier, as recited by the claims of Group II.

As correctly pointed out by the Examiner, Mohammed has not explicitly taught floating point registers. Accordingly, the Examiner cites Blomgren. In contrast to the above-recited features of the claims of Group II, as well as Mohammed, Blomgren teaches a mechanism for rapid reconfiguration of pipeline alignment between a pipeline optimized for risk instructions and one optimized for CISC instructions with the uses of MUXes and mode registers. (*See*, Abstract, and also as depicted with reference to Blomgren's FIG. 2.)

According to the Examiner:

> it would have been obvious to one of ordinary skill in the art at the time the invention was made to use the shared floating point registers in Blomgren in the device of Mohammed to increase compatibility between data types, decrease size and the cost of processor and increase processor efficiency. (pg. 10 of Office Action mailed June 30, 2004.)

Blomgren teaches the detection of an emulated CISC instruction and switches to a RISC mode when such an instruction is detected. The switch to the RISC mode results in the loading of various RISC instructions to perform the CISC instruction. However, the entire text of Blomgren is devoid of any reference to detecting whether an input from a plurality of floating point registers includes a token.

Furthermore, Blomgren is devoid of any teachings of performing token specific processing if the processor is in a second mode or processing the input to render an arithmetic result if the processor is in a first mode. In contrast, Blomgren teaches a CISC instruction decoder that detects emulated instructions and signals the unknown opcode, which the Examiner equates to a token, to load pointers with the address of an emulation routine. (*See* cols. 6-7, lines 61-10.)

Accordingly, although Blomgren describes emulation of certain CISC routines with RISC instructions, the processing performed within Blomgren does not vary according to a first or second mode of a multi-mode processor. Accordingly, the combination of Mohammed in view of Blomgren fails to teach or suggest the recited features of the claims of Group II.

Furthermore, although Mohammed teaches processing of short format and long format instructions, Mohammed provides no teachings or suggestions with regards to first and second ISAs to share the floating point registers. In fact, the dynamic reconfiguration capability of the functional units enables Mohammed to provide a single ISA for a hybrid VLIW-SIMD programming model for instruction packets having different bit lengths. Hence, the teachings or Mohammed are specifically limited to a single ISA, which uses a mode field within received instruction packets to determine whether instructions within received packets are a long or short format. Based on the instruction format, the functional units are dynamically reconfigured to execute such instructions.

Therefore, since Mohammed is specifically limited to a single ISA, Mohammed cannot be modified according to Blomgren to share floating point registers between first and second ISAs, as recited by the claims of Group II. Furthermore, Applicants respectfully submit that the Examiner fails to illustrate a teaching or suggestion for combining the teachings of Mohammed in view of Blomgren.

Applicants submit that there is no suggestion as to any variation in the word sizes between the RISC and CISC instruction sets within Blomgren. (*See*, col. 6, lines 37-44.) Hence, the hybrid ISA, as taught by Mohammed, for executing variable length instructions has virtually no relation to the emulation of complex CISC instructions using a plurality of RISC instructions or emulated instructions, as referred to by Blomgren. Accordingly, Applicants respectfully submit that the features of the claims of Group II could only be arrived at through inappropriate hindsight.

Therefore, Applicants respectfully submit that a *prima facie* case of obviousness of the claims of Group I is not established and therefore the rejection of Claims 1, 2, 4, 5, 7, 8 and 18 is erroneous and should be overturned. Accordingly, Applicants respectfully request that the §103(a) rejection of the claims of Group II be overturned.

D. Group III: Rejection of Claim 3 As Obvious Over Mohammed in View of Blomgren and Further in View of Loper

The Examiner rejected all pending claims, including Claim 3 of Group III under 35 U.S.C. §103(a) as obvious over Mohammed in view of Blomgren and further in view of Loper.

1. Errors of Law and Fact in the Rejection

The Examiner has made the same errors as previously described with respect to the rejected claims of Group II. In addition, the Examiner has failed to show that the prior art references of Mohammed in view of Blomgren and further in view of Loper teach or suggest all the features of Claim 3 of Group III.

The combination of Mohammed in view of Blomgren and further in view of Loper fails to rectify the deficiencies attributed to the combination of Mohammed in view of Blomgren to teach or suggest a processor mode identifier. As indicated above with reference to Group II, the teachings of Mohammed are specifically limited to a single ISA, which uses a mode field to direct the scheduler to schedule instructions for execution within dynamically reconfigurable functional units. As a result, Mohammed cannot be modified, as suggested by the Examiner, to share floating point registers between first and second ISAs, as recited by Claim 3, since Mohammed is limited to a single ISA. Therefore, Applicants respectfully submit that the features of the claims of Group III could only be arrived at through inappropriate hindsight.

It is well established that obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention absent the teaching or suggestion

supporting such combination. ACS Hospital Sys., Inc. v. Montefiore Hospital, 732 F.2d. 1572, 1577, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984). Also, one cannot find obviousness through hindsight to construct a claimed invention from elements of the prior art. In re Warner, 379 F.2d 1011, 1016, 154 U.S.P.Q. 173, 177 (C.C.P.A. 1967).

Hence, a *prima facie* case of obviousness of the claims of Group III has not been established and the rejection of Claim 3 is therefore erroneous.

2. Specific Limitations Not Described in the Prior Art

Claim 3 of Group III requires:

> pre-processing hardware to detect if a token exists in the input;
> an arithmetic unit responsive to the input and the mode identifier; and
> post-processing hardware to perform a token specific operation if a token exists in the input.

3. Explanation Why Such Limitations Render the Claims Non-obvious over the Prior Art

The Applicants claim a processor including pre-processing and post-processing hardware to process an input to render an arithmetic result if a processor is in a first mode and performing a token specific operation if a processor is in a second mode when a token is detected as an input. In contrast, the teachings of Loper are specifically limited to either requesting data for a speculative load instruction or immediately suspending the speculative load instruction according to the state of a bit in the ESA table as shown in FIGS. 2A and 2B associated with the speculative load instruction. (*See*, col. 2, lines 2-23.)

Although Blomgren describes emulation of certain CISC routines with RISC instructions, the processing performed within Blomgren does not vary according to a processor mode as indicated by a mode identifier. Likewise, both Loper and Mohammed vary operation according to an instruction state. Applicants respectfully submit that the varying of operation according to an instruction state does not teach the pre-processing and post-processing hardware of a token according to a processor mode, as required by Claim 3 of Group III.

Furthermore, Applicants respectfully submit that since Mohammed is limited to a single ISA, no combination of references, including Blomgren, Loper or the references of record, can be used to modify Mohammed to share floating point registers between plural ISAs. Accordingly, Applicants respectfully submit that the features of the claims of Group III could only be arrived at through inappropriate hindsight.

Therefore, a *prima facie* case of anticipation of the claims of Group III is not established and the rejection of Claim 3 should be overturned. Accordingly, Applicants respectfully request that the §102(e) rejection of Claim 3 of Group III be overturned.

E. Group IV: Rejection of Claim 6 As Obvious Over Mohammed in View of Blomgren and Further in View of Loper

The Examiner rejected all pending claims, including Claim 6 of Group IV under 35 U.S.C. §103(a) as obvious over Mohammed in view of Blomgren and further in view of Loper.

1. Errors of Law and Fact in the Rejection

The Examiner has made the same errors as previously described with respect to the rejected claims of Groups II and III. In addition, the Examiner has failed to show that the prior art references of Mohammed in view of Blomgren and further in view of Loper teach or suggest all the features of Claim 6 of Group IV.

Applicants respectfully submit that since Mohammed is limited to a single ISA, no combination of references, including Blomgren, Loper or the references of record, can be used to modify Mohammed to share floating point registers between a first ISA and a second ISA since Mohammed is limited to the single ISA. Accordingly, Applicants respectfully submit that the features of the claims of Group IV could only be arrived at through inappropriate hindsight.

Furthermore, Applicants submit that the combination of references provides no reference to a token that represents a "not a thing value" (NaTVal) defines an unsuccessful speculative load request, as required by Claim 6 of Group IV. Hence, a *prima facie* case of obviousness of Claim 6 of Group IV has not been established and the rejection of Claim 6 is therefore erroneous.

2. Specific Limitations Not Described in the Prior Art

Claim 6 of Group IV requires:

> wherein the token represents a "not a thing value" (NaTVal) that defines an unsuccessful speculative load request.

3. Explanation Why Such Limitations Render the Claims Non-obvious over the Prior Art

The Applicants claim a processor including pre-processing and post-processing hardware to perform a token specific operation if a processor is in a second mode, when a token is detected, wherein a token representing a "not a thing value" (NaTVal) defines an unsuccessful speculative load request. In contrast, the teachings of Loper are specifically limited to either executing or immediately suspending a speculative load instruction according to the state of a bit in an ESA table (*See*, FIGS. 2A and 2B) associated with the speculative load instruction.

Accordingly, although Blomgren describes emulation of certain CISC routines with RISC instructions, the processing performed within Blomgren does not vary according to a processor mode as indicated by a mode identifier. Likewise, both Loper and Mohammed vary operation according to an instruction state.

Applicants respectfully submit that the detection of emulated instructions does not teach NaTVal tokens to indicate an unsuccessful speculative load request, as required by Claim 6 of Group IV.

Furthermore, Applicants respectfully submit that since Mohammed is limited to a single ISA, no combination of references, including Blomgren, Loper or the references of record, can be used to modify Mohammed to share floating point registers between its single ISA. Accordingly, Applicants respectfully submit that the features of the claims of Group III could only be arrived at through inappropriate hindsight.

Therefore, a *prima facie* case of obviousness of Claim 6 of Group IV is not established and the rejection of Claim 6 should be overturned. Accordingly, Applicants respectfully request that the §103(a) rejection of Claim 6 of Group III be overturned.
be overturned.

F. Group V: Rejection of Claim 9 as Obvious over Mohammed in View of Blomgren

The Examiner rejected all pending claims, including Claim 9 of Group V under 103(a) as obvious over Mohammed in view of Blomgren.

1. Errors of Law and Fact in the Rejection

The Examiner has made the same errors as described previously with respect to the rejected claims of Groups II-IV. In addition, the Examiner has failed to show that the prior art references of Mohammed in view of Blomgren teach or suggest all claim features of Claim 9 of Group V.

As indicated above with reference to Group II, the teachings of Mohammed are specifically limited to a single ISA, which uses an instruction state to direct the scheduler to schedule instructions for execution within dynamically reconfigurable functional units. Consequently, Mohammed, as well as the references of record, fail to teach a mode identifier to indicate a 32-bit word ISA mode and a 64-bit word ISA mode, as recited by Claim 9, since Mohammed is limited to a single ISA.

2. Specific Limitations Not Described in the Prior Art

Claim 9 of Group V requires a 32 bit word ISA mode and a 64 bit word ISA mode, which as noted by the Examiner is not taught or suggested by Blomgren.

3. Explanation Why Such Limitations Render the Claims Non-obvious over the Prior Art

As indicated above with reference to Group V, the teachings of Mohammed are specifically limited to a single ISA, which uses an instruction state to direct the scheduler to schedule instructions for execution within dynamically reconfigurable functional units. As a result, Mohammed, as well as the references of record, fail to teach a mode identifier to indicate a 32-bit

word ISA mode and a 64-bit word ISA mode, as recited by Claim 9. Accordingly, Applicants respectfully submit that the features of the claims of Group V could only be arrived at through inappropriate hindsight.

Therefore, Applicants respectfully submit that a *prima facie* case of obviousness of Claim 9 of Group V is not established and therefore the rejection of Claim 9 is erroneous and should be overturned. Accordingly, Applicants respectfully request that the §103(a) rejection of Claim 9 of Group V be overturned.

G. Group VI: Rejection of Claims 12, 13, 15, 16 and 19 As Obvious Over Mohammed in View of Blomgren

The Examiner rejected all pending claims, including Claims 12, 13, 15, 16 and 19 of Group VI under 35 U.S.C. §103(a) as being obvious over Mohammed in view of Blomgren.

1. Errors of Law and Fact in the Rejection

The Examiner has made the same errors as described previously with respect to the rejected claims of Groups II-V. In addition, the Examiner has failed to show that the prior art references of Mohammed in view of Blomgren teach or suggest all claim features of Claims 12, 13, 15, 16 and 19 of Group VI. The Federal Circuit Court of Appeals in In re Rijckaert, 9 F.3d 1531, 28 U.S.P.Q. 2d 1955 (Fed. Cir. 1993) held that:

> In rejecting claims under 35 U.S.C. § 103, the examiner bears the initial burden of presenting a *prima facie* case of obviousness. . . . "A *prima facie* case of obviousness is established when the teaching from the prior art itself would appear to have suggested the claimed subject matter to a person of ordinary skill in the art.". . . If the examiner fails to establish a *prima facie* case, the rejection is improper and will be overturned. (Emphasis added.) 9 F.3d at 1532, 28 U.S.P.Q. 2d at 1956.

The Examiner has rejected Claims 12, 13, 15, 16 and 19 of Group VI under 35 U.S.C. §103(a) as being unpatentable over Mohammed in view of Blomgren. The Examiner cites Blomgren, which according to the Examiner teaches shared floating point register. According to the Examiner:

> It would be obvious to a person of ordinary skill in the art at the time the invention was made to use the shared floating-point registers of Blomgren in the device of Mohammed to increase compatibility between data types, decrease size and cost of the processor, and increase processor efficiency. (*See* pg. 10, ¶ 23 of the Office Action mailed June 30, 2004.)

However, even if Blomgren disclosed shared floating point registers between ISAs, the Examiner fails to teach that it would be obvious to combine the missing elements provided by Blomgren within the teachings of Mohammed.

Hence, a *prima facie* case of obviousness of Claims 12, 13, 15, 16 and 19 of Group VI have not been established and the rejection of Claims 12, 13, 15, 16 and 19 is erroneous and should be overturned.

2. Specific Limitations Not Described in the Prior Art

Each of the independent claims of Group VI requires:

> fetching an input from at least one of a plurality of floating-point registers;
> detecting whether the input includes a token;
> if the token is detected in the input, checking what mode the processor is in;
> if the processor is in a first mode, processing the input to render an arithmetic result operation;
> if the processor is in a second mode, performing a token specific operation. (Emphasis added.)

3. Explanation Why Such Limitations Render the Claims Non-obvious over the Prior Art

The Applicants claim a method for processing an input to render an arithmetic result if a processor is in a first mode and performing a token specific operation if a processor is in a second mode when a token is detected as an input. In contrast, Mohammed describes a mode field included as a part of the 256-bit instruction packet, wherein a first bit value can be set to identify a long format of the instruction and a second bit value can be set to identify a short form instruction. (*See*, col. 6, lines 21-29.) According to the Examiner, the mode field, as taught by Mohammed, teaches the detection of whether an input includes a token.

Applicants respectfully submit that the instruction packet and corresponding mode field as taught by Mohammed do not teach or suggest the detection of whether a token is received as an input. In other words, Applicants respectfully submit that the inclusion of mode bits within instructions packets, as taught by Mohammed, to indicate whether each instruction contained in the packet is in a long format or short formal fails to teach or suggest the detection of whether an input contains a token, as recited by the claims of Group VI.

Furthermore, Applicants respectfully submit that the mode field within the instruction packets, as taught by Mohammed, identifies instructions as having a long format and short format; it is therefore limited to describing an instruction state. Conversely, the claims of Group IV recite checking what mode a processor is in if a token is detected in the input. The mode referred is a processor mode and not a state of the instructions executed by the processor. Accordingly, Mohammed fails to teach this above-recited feature of the claims of Group VI.

In addition, the claims of Group VI recite processing of an input to render an arithmetic result or performing a token specific operation based on a processor mode. Applicants respectfully submit the association of a bit within a mode field of an instruction packet, as taught by

Mohammed to identify instructions is either a long format or a short format, does not teach or suggest the above-recited features of the claims of Group VI.

Furthermore, the Examiner's citing of Blomgren also fails to rectify the above-described deficiencies attributed to Mohammed. The emulation of complex CISC instructions using a plurality of RISC instructions as taught by Blomgren, fails to at least teach or suggest the processing of an input to render an arithmetic result or performing a token specific operation based on a processor mode, as recited by the claims of Group VI. (*See*, col. 7, lines 1-10.) Hence, Applicants respectfully submit that the teachings of Mohammed, in combination with the teachings of Blomgren, fail to at least teach or suggest each of the claim features recited by the claims of Group VI.

Moreover, Applicants respectfully submit that the Examiner fails to illustrate some suggestion or motivation to modify Mohammed in view of Blomgren. Applicants respectfully submit that the hybrid ISA as taught by Mohammed for executing variable lengths instructions has virtually no relation to the emulation of complex CISC instructions using a plurality of RISC instructions or emulated instructions, as referred to by Blomgren. Accordingly, Applicants respectfully submit that the features of the claims of Group VI could only be arrived at through inappropriate hindsight.

Therefore, a *prima facie* case of obviousness of the claims of Group VI is not established and the rejection of Claims 12, 13, 15, 16 and 19 should be overturned. Accordingly, Applicants respectfully request that the §103(a) rejection of the claims of Group VI be overturned.

H. Group VII: Rejections of Claim 10

1. Rejection of Claim 10 As Obvious Over Mohammed in View of Blomgren

a. Errors of Law and Fact in the Rejection

The Examiner has made the same errors as described previously with respect to the rejected claims of Groups II-VI. In addition, the Examiner has failed to show that the prior art references of Mohammed in view of Blomgren teach or suggest all claim features of Claim 10 of Group VII.

The Examiner has rejected Claim 10 of Group VII under 35 U.S.C. §103(a) as being unpatentable over Mohammed in view of Blomgren. The Examiner cites Blomgren, which according to the Examiner teaches shared floating point register. According to the Examiner:

> It would be obvious to a person of ordinary skill in the art at the time the invention was made to use the shared floating-point registers of Blomgren in the device of Mohammed to increase compatibility between data types, decrease size and cost of the processor, and increase processor efficiency. (*See* pg. 10, ¶ 23 of the Office Action mailed June 30, 2004.)

However, even if Blomgren disclosed shared floating point registers between ISAs, the Examiner fails to teach that it would be obvious to combine the missing elements provided by Blomgren within the teachings of Mohammed. Accordingly, Applicants respectfully submit that the features of the claims of Group VII could only be arrived at through inappropriate hindsight.

b. Specific Limitations Not Described in the Prior Art

Claim 10 requires:

fetching an input from at least one of a plurality of floating-point registers;
detecting whether the input includes a token;
if the token is detected in the input, checking what mode the processor is in;
if the processor is in a first mode, processing the input to render an arithmetic result operation;
if the processor is in a second mode, performing a token specific operation. (Emphasis added.)

c. Explanation Why Such Limitations Render the Claims Non-obvious over the Prior Art

Applicants respectfully submit that the instruction packet and corresponding mode field as taught by Mohammed do not teach or suggest the detection of whether a token is received as an input. In other words, Applicants respectfully submit that the inclusion of mode bits within instructions packets, as taught by Mohammed, to indicate whether each instruction contained in the packet is in a long format or short formal fails to teach or suggest the detection of whether an input contains a token, as recited by the claims of Group VII.

Furthermore, Applicants respectfully submit that the mode field within the instruction packets, as taught by Mohammed, identifies instructions as having a long format and short format; it is therefore limited to describing an instruction state. Conversely, the claims of Group VII recite checking what mode a processor is in if a token is detected in the input. The mode referred is a processor mode and not a state of the instructions executed by the processor. Accordingly, Mohammed fails to teach this above-recited feature of the claims of Group VII.

In addition, the claims of Group VII recite processing of an input to render an arithmetic result or performing a token specific operation based on a processor mode. Applicants respectfully submit the association of a bit within a mode field of an instruction packet, as taught by Mohammed to identify instructions is either a long format or a short format, does not teach or suggest the above-recited features of the claims of Group VII.

2. Rejection of Claim 10 As Anticipated by Loper

The Examiner rejected Claim 10 of Group VII under 35 U.S.C. §102(b) as being anticipated by Loper.

a. Errors of Law and Fact in the Rejection

Applicants respectfully assert that the Examiner has failed to adequately set forth a *prima facie* rejection under 35 U.S.C. §102(b). "Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, *arranged as in the claim.*" *Lindemann Maschinenfabrik v. American Hoist & Derrick* ("Lindemann"), *730 F.2d 452, 1458 (Fed. Cir. 1994)*(emphasis added). Additionally, each and every element of the claim must be exactly disclosed in the anticipatory reference. *Titanium Metals Corp. of American v. Banner* ("Banner Titanium"), *778 F.2d 775, 777 (Fed. Cir. 1985).* In addition, the Examiner has failed to show that each and every element of the claims of Group IV are exactly disclosed by Blomgren. Banner Titanium. *Id.*

According to the Examiner, executing or immediately suspending a speculative load instruction according to the state of a bit in an ESA associated with a speculative load instruction anticipates the claims of Group VII. Applicants respectfully submit that the Examiner has incorrectly associated an instruction state with a processor mode and a bit of the ESA table of Loper with the token as recited by the claims of Group VII. (*See* cols. 6-7, lines 61–10.)

The teachings of Loper are strictly directed to prohibiting the displacement of:

> essential data stored in a data cache by some irrelevant data obtained from the system bus because of a wrongful execution of a speculative load instruction caused by misdirection. (*See*, col. 1, lines 44-47.)

Applicants respectfully submit that the bit associated with the speculative load instructions and accessing of such bit to determine whether to execute a speculative load instruction or suspend the speculative load instruction does not teach or suggest the detection of whether an input contains a token or processing the token according to a processor mode, as recited by the claims of Group VII. Hence, a *prima facie* case of anticipation of the claims of Group VII has not been established and the rejection of Claims 10 and 14 is therefore erroneous.

b. Specific Limitations Not Described in the Prior Art

Claim 10 requires:

> fetching an input from at least one of a plurality of floating-point registers;
> detecting whether the input includes a token;
> if the token is detected in the input, checking what mode the processor is in;
> if the processor is in a first mode, processing the input to render an arithmetic result operation;
> if the processor is in a second mode, performing a token specific operation. (Emphasis added.)

c. Explanation Why Such Limitations Render the Claims Unanticipated by the Prior Art

The teachings of Loper are strictly directed to prohibiting the displacement of:

> essential data stored in a data cache by some irrelevant data obtained from the system bus because of a wrongful execution of a speculative load instruction caused by misprediction. (*See*, col. 1, lines 44-47.)

Furthermore, Applicants respectfully submit that the state of a bit associated with the speculative load instructions within an ESA table as taught by Loper provides no teachings or suggestions with regards to a processor mode. Applicants respectfully submit that the processor mode, as recited by the claims of Group VII, refers to what may be referred to as a "global processor mode." Conversely, the bits associated with speculative load instructions simply refer to an instruction state, which is specific to the respective speculative load instruction and does not coincide with a processor mode, as recited by Group VII. However, the case law is quite clear in establishing that each and every element of the claim must be exactly disclosed in the anticipatory reference. Banner Titanium, *Id.*

Therefore, a *prima facie* case of anticipation of Claim 10 of Group VII is not established and the rejection of Claim 10 should be overturned. Accordingly, Applicants respectfully request that the §102(b) rejection of Claim 10 of Group VII be overturned.

I. Group VIII: Rejection of Claim 14 as Anticipated by Loper

The Examiner rejected Claim 14 of Group VIII under 35 U.S.C. §102(b) as being anticipated by Loper.

1. Errors of Law and Fact in the Rejection

The Examiner has made the same errors as previously described with reference to the rejected claims of Group VII. In addition, the Examiner has failed to show that each and every element of the claim of Group VIII is exactly disclosed by Loper.

According to the Examiner, the executing or immediately suspending the speculative load instruction according to the state of a bit in the ESA table as shown in FIGS. 2A and 2B associated with a speculative load instruction anticipates the claims of Group VIII. Applicants respectfully submit that the Examiner has incorrectly associated an instruction state with a processor mode and a bit of the ESA table of Loper with the token as recited by the claims of Group VIII. (*See* cols. 6-7, lines 61–10.)

The teachings of Loper are strictly directed to prohibiting the displacement of:

> essential data stored in a data cache by some irrelevant data obtained from the system bus because of a wrongful execution of a speculative load instruction caused by misdirection. (*See*, col. 1, lines 44-47.)

Applicants respectfully submit that the bit associated with the speculative load instructions and accessing of such bit to determine whether to execute a speculative load instruction or suspend the speculative load instruction does not teach or suggest the detection of whether an input contains a token or processing the token according to a processor mode, as recited by the claims of Group VIII. Hence, a *prima facie* case of anticipation of the claims of Group VIII has not been established and the rejection of Claim 14 is therefore erroneous.

2. Specific Limitations Not Described in the Prior Art

The claim of Group VIII requires:

> wherein the token represents a "not a thing value" (NaTVal) that defines an unsuccessful speculative load request.

3. Explanation Why Such Limitations Render the Claims Unanticipated by the Prior Art

The teachings of Loper are strictly directed to prohibiting the displacement of:

> essential data stored in a data cache by some irrelevant data obtained from the system bus because of a wrongful execution of a speculative load instruction caused by misprediction. (*See*, col. 1, lines 44-47.)

Furthermore, Applicants respectfully submit that the state of a bit associated with the speculative load instructions within an ESA table as taught by Loper provides no teachings or suggestions with regards to a processor mode. Applicants respectfully submit that the processor mode, as recited by the claims of Group VIII, refers to what may be referred to as a "global processor mode." Conversely, the bits associated with speculative load instructions simply refer to an instruction state, which is specific to the respective speculative load instruction and does not coincide with a processor mode, as recited by Group VIII.

Moreover, the processing of an input to render an arithmetic result or to perform a token-specific operation is neither taught nor suggested by the execution or suspension of a speculative load instruction according to a state of a bit associated with the speculative load instruction. Furthermore, Applicants respectfully submit that the execution/suspension of speculative load instructions does not teach the detection of NaTVal token that defines an unsuccessful speculative load request, as required by Claim 14 of Group VIII. However, the case law is quite clear in establishing that each and every element of the claim must be exactly disclosed in the anticipatory reference. Banner Titanium, *Id.*

Therefore, a *prima facie* case of anticipation of Claim 14 of Group VIII is not established and the rejection of Claim 14 should be overturned. Accordingly, Applicants respectfully request that the §102(b) rejection of Claim 14 of Group VIII be overturned.

J. Group IX: Rejection of Claim 17 as Obvious over Mohammed in View of Blomgren

The Examiner rejected all pending claims, including Claim 17 under 103(a) as obvious over Mohammed in view of Blomgren.

1. Errors of Law and Fact in the Rejection

The Examiner has made the same errors as described previously with respect to the rejected claims of Groups II-VI. In addition, the Examiner has failed to show that the prior art references of Mohammed in view of Blomgren teach or suggest all claim features of the claims of Group VII.

As indicated above with reference to Group II, the teachings of Mohammed are specifically limited to a single ISA, which uses a mode field to direct the scheduler to schedule instructions for execution within dynamically reconfigurable functional units. As a result, Mohammed, as well as the references of record, fail to teach or suggest a mode identifier to indicate a 32-bit word ISA mode and a 64-bit word ISA mode, as recited by Claim 17, since Mohammed is limited to a single ISA.

2. Specific Limitations Not Described in the Prior Art

Claim 17 of Group IX requires a 32 bit size ISA mode and a 64 bit word ISA mode, which as noted by the Examiner is not taught or suggested by Blomgren.

3. Explanation Why Such Limitations Render the Claims Non-Obvious over the Prior Art

Mohammed, as well as the references of record, fail to teach or suggest a mode identifier to indicate a 32-bit word ISA mode and a 64-bit word ISA mode, as recited by Claim 17, since Mohammed is limited to a single ISA. Therefore, Applicants respectfully submit that the features of the claims of Group IX could only be arrived at through inappropriate hindsight.

Therefore, Applicants respectfully submit that a *prima facie* case of obviousness of Claim 17 of Group IX is not established and therefore the rejection of Claim 17 is erroneous and should be overturned. Accordingly, Applicants respectfully request that the §103(a) rejection of Claim 17 of Group IX be overturned.

IX. CONCLUSION AND RELIEF

Based on the foregoing, Applicant requests that the Board overturn the rejection of all pending claims and hold that all of the claims of the present application are allowable.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Dated: August 31, 2004

Joseph Latz, Reg. No. 43,765

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025
(310) 207-3800

CERTIFICATE OF MAILING:
I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, with sufficient postage, in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on August 31, 2004

Marilyn Bass August 31, 2004

X. APPENDIX

The claims involved in this Appeal are as follows:

1. (Previously Presented) A processor comprising:
a first instruction set engine to process instructions from a first instruction set architecture (ISA) having a first word size;

a second instruction set engine to process instructions from a second ISA having a second word size, the second word size being different than the first word size;
a mode identifier;

a plurality of floating-point registers shared by the first instruction set engine and the second instruction set engine; and

a floating-point unit coupled to the floating-point registers, the floating-point unit processing an input responsive to the mode identifier to produce an output.

2. (Original) The processor of Claim 1 wherein the mode identifier is one of a plurality of bits in a processor status register.

3. (Previously Presented) The processor of Claim 1 wherein the floating-point unit comprises:

pre-processing hardware to detect if a token exists in the input;

an arithmetic unit responsive to the input and the mode identifier; and

post-processing hardware to perform a token specific operation if a token exists in the input.

4. (Previously Presented) The processor of Claim 1 wherein the input includes data stored in at least one of the floating-point registers.

5. (Previously Presented) The processor of Claim 1 wherein the input may contain a token, wherein the floating-point registers are 82 bits wide, and wherein the token being an 82 bit processor known value.

6. (Previously Presented) The processor of Claim 3 wherein the token represents a "not a thing value" (NaTVal) that defines an unsuccessful speculative load request.

7. (Original) The processor of Claim 1 wherein the floating point registers each comprise:

a sign bit,

an exponent; and

a significand.

8. (Original) The processor of Claim 1 wherein the mode identifier indicates whether the processor is in a first mode or a second mode.

9. (Previously Presented) The processor of Claim 1 wherein the mode identifier indicates whether the processor is in a 32 bit word ISA mode or a 64 bit word ISA mode.

10. (Previously Presented) A method in a processor comprising:

fetching an input from at least one of a plurality of floating-point registers;

detecting whether the input includes a token;

if the token is detected in the input, checking what mode the processor is in;

if the processor is in a first mode, processing the input to render an arithmetic result;

if the processor is in a second mode, performing a token specific operation; and

producing an output.

11. (Previously Presented) The method of Claim 10 wherein the input is comprised of at least one operand and at least one operator; wherein detecting comprises examining the at least one operand to determine whether any of the operands correspond to the token; and wherein checking comprises examining a mode identifier to determine whether the processor is in the first mode or the second mode.

12. (Previously Presented) The method of Claim 10 wherein processing comprises executing at least one operation on the at least one operand according to the at least one operator to achieve a result.

13. (Original) The method of Claim 10 wherein performing comprises propagating the token; and wherein producing output comprises setting the output to be the token.

14. (Original) The method of Claim 10 wherein the token represents a "not a thing value" (NaTVal) that defines an unsuccessful speculative load request.

15. (Original) The method of Claim 10 wherein checking comprises checking a mode identifier.

16. (Original) The method of Claim 10 wherein checking comprises checking a mode identifier bit in a processor status register.

17. (Original) The method of Claim 11 wherein the first mode is a 32 bit word ISA mode and the second mode is a 64 bit word ISA mode.

18. (Previously Presented) A multi-mode processor comprising:

a plurality of instruction set engines to process instructions from a plurality of instruction set architectures having different word sizes;

a mode identifier;

a plurality of floating-point registers shared by the instruction set engines; and

a plurality of floating-point units coupled to the floating-point registers, the floating-point units processing an input responsive to the mode identifier.

19. (Previously Presented) A method in a multi-mode processor comprising:

fetching an input from at least one of a plurality of floating-point registers;

detecting whether the input includes at least one token of a plurality of tokens;

if at least one token is detected in the input, checking what mode the processor is in;

processing the input to render an arithmetic result when the processor is in at least a first mode of a plurality of modes; and

performing a token specific operation when the processor is in at least a second mode of a plurality of modes.